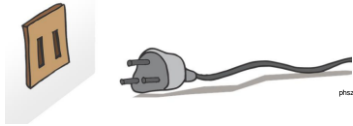
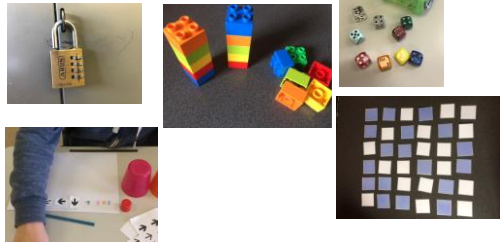




Informatique sans électricité



Concepts de base de l'informatique



Des unités sur les nombres binaires, les systèmes de filtrage et de tri ainsi que les systèmes de cryptage et les programmes/instructions transmettent la connaissance et la compréhension de la fonction de l'ordinateur.

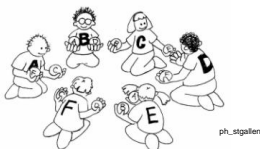


Concepts informatiques basé sur le jeu comme principe d'apprentissage

matériel quotidien



Tâches avec des objets du quotidien (stylos, papier, cordes, cartes à jouer, rouleaux de papier, dés, etc.)



Les élèves doivent communiquer, résoudre des problèmes, faire preuve de créativité, utiliser leurs connaissances de manière adéquate.



Les unités sont durables (pas de problèmes de détails techniques)



Rentable même avec des ressources financières limitées (Döbeli, 2007) et rapidement réalisable.

2 | Informatik

1. Die Schülerinnen und Schüler können Daten aus ihrer Umwelt darstellen, strukturieren und auswerten.
2. Die Schülerinnen und Schüler können einfache Problemstellungen analysieren, mögliche Lösungsverfahren beschreiben und in Programmen umsetzen.
3. Die Schülerinnen und Schüler verstehen Aufbau und Funktionsweise von informationsverarbeitenden Systemen und können Konzepte der sicheren Datenverarbeitung anwenden.

1 Die Schülerinnen und Schüler können Daten aus ihrer Umwelt darstellen, strukturieren und auswerten.

Die Schülerinnen und Schüler ...

- 1 können Dinge nach selbst gewählten Eigenschaften ordnen, damit sie ein Objekt mit einer bestimmten Eigenschaft schneller finden (z.B. Farbe, Form, Größe).
- 2 können unterschiedliche Darstellungsformen für Daten verwenden (z.B. Symbole, Tabellen, Diagramm).
- 3 können Daten mittels selbstbestimmter Darstellungsform veranschaulichen.
- 4 können analoge und digitale Darstellungen von Daten (Text, Zahl, Bild und Ton) und können die entsprechenden Darstellungsformen zuordnen.
- 5 können die Backvorgänge der von ihnen gewählten Dokumententypen erkennen und verändern (z.B. Dokumententypen auf dem Computer: Dokument, Formular, Textdokument).
- 6 verstehen die Zusammenhänge von Netzwerkelementen und können komplexe Daten darstellen.
- 7 können Dokumente so ablegen, dass auch andere sie wieder finden können (z.B. können digitale Dokumente verschlüsselt und/oder nicht).
- 8 können Daten in einer Standardsprache erstellen, austauschen und automatisch auswerten.
- 9 können Methoden zur Datenverarbeitung unterscheiden und anwenden (Backup, Synchronisation, Vervielfältigung).

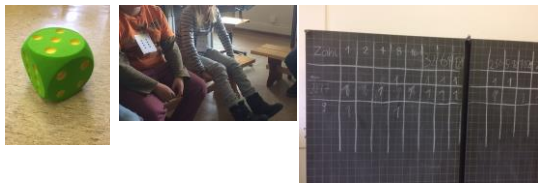
2 Die Schülerinnen und Schüler können einfache Problemstellungen analysieren, mögliche Lösungsverfahren beschreiben und in Programmen umsetzen.

Die Schülerinnen und Schüler ...

- 1 können formale Anleitungen erkennen und ihnen folgen (z.B. Koch- und Backrezepte, Spiel- und Bastelanleitungen, Tanzchoreographien).
- 2 können durch Probieren Lösungswege für einfache Problemstellungen aufbauen und auf Formeln übertragen (z.B. einen Weg beschreiben, eine Schaltung angeben). Ein kleiner, selbstbestimmter Übertragungsweg ist möglich.
- 3 können Details mit Symbolen und Notationen aus ihrer Umwelt erkennen, beschreiben und austauschen (z.B. mittels Klappergitter).
- 4 können einfache Details mit Symbolen, bedingten Anweisungen und Parametern beschreiben und austauschen.
- 5 verstehen, dass ein Computer nur selbst definierte Anweisungen ausführen kann und dass ein Programm eine Abfolge von solchen Anweisungen ist.
- 6 können Programme mit Symbolen, bedingten Anweisungen und Parametern schreiben und ausführen.
- 7 können selbstbestimmte Übertragewege für einfache Probleme in Form von Aufträgen und formaler Computerprogrammen mit Symbolen, bedingten Anweisungen und Parametern beschreiben.
- 8 können selbstbestimmte Algorithmen in Form von Aufträgen und formaler Computerprogrammen mit Symbolen und Parametern beschreiben.
- 9 können abschließende Algorithmen zur Lösung bestehender Probleme erkennen und beschreiben (z.B. Umkehr- und Umkehrfunktion, Sortieralgorithmus).

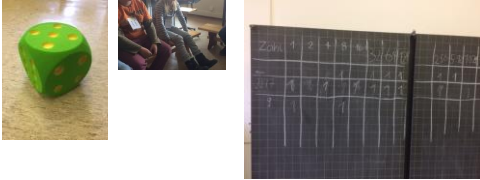
Datenstrukturen	Algorithmen	Informationssysteme	Darstellungskompetenz „Probleme lösen“
1) Schülerinnen und Schüler können Daten aus ihrer Umwelt darstellen, strukturieren und auswerten sowie durch Vorschlagslösung schützen.	2) Schülerinnen und Schüler können Problemstellungen analysieren und selbst entdeckte Lösungsverfahren beschreiben sowie in Programmen umsetzen.	3) Die Schülerinnen und Schüler verstehen Aufbau und Funktionsweise von informationsverarbeitenden Systemen und können Konzepte der sicheren Datenverarbeitung anwenden.	Probleme lösen (Inhalts-, Strukturierungs-, Formulierungs- und Zielsetzungsprobleme)
a) können Dinge nach selbst gewählten Eigenschaften ordnen, damit sie ein Objekt mit einer bestimmten Eigenschaft schneller finden (z.B. Farbe, Form, Größe).	a) können formale Anleitungen erkennen und ihnen folgen (z.B. Koch- und Backrezepte, Spiel- und Bastelanleitungen, Tanzchoreographien).	a) können Geräte ein- und ausschalten, Programme starten, betreiben und beenden sowie einfache Funktionen nutzen. b) können sich mit eigenem Logon in einem lokalen Netzwerk oder einer Lernumgebung anmelden. c) können Dokumente selbstständig ablegen und wieder finden. d) können mit grundlegenden Elementen der Bedienoberfläche umgehen (Fenster, Menü, mehrere geöffnete Programme).	(0) Erkennt ein Problem und kann dieses formulieren. (1) Kann sich mit anderen zu einem Problem austauschen. (2) Kann Probleme basierend auf Hintergründen erklären und diskutieren.

Codes - Codes binaires



Codes – Binärcodes

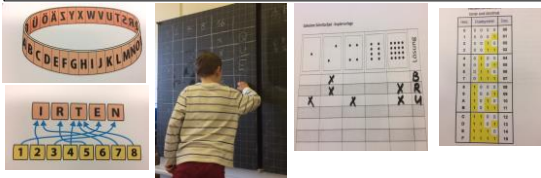
- Les ordinateurs stockent les informations dans des codes binaires. Binaire, puisque deux états 0 - 1.
- Les processeurs (hardware) pour l'addition et la multiplication de nombres binaires peuvent être construits plus facilement que pour les nombres décimaux.



Computational thinking 13

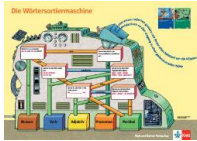
Codage / Décodage / Chiffrement / Déchiffrement / Écriture secrète

- Nous utilisons une clé numérique pour crypter un texte selon des règles fixes (cryptage). Après ce processus, le texte crypté peut être envoyé sur un canal non sécurisé.
- La partie de l'informatique qui traite du cryptage et du décryptage des données est appelée cryptologie.



Computational thinking 14

Tri / Regroupement / Traitement de l'information



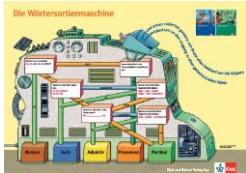
Computational thinking 15



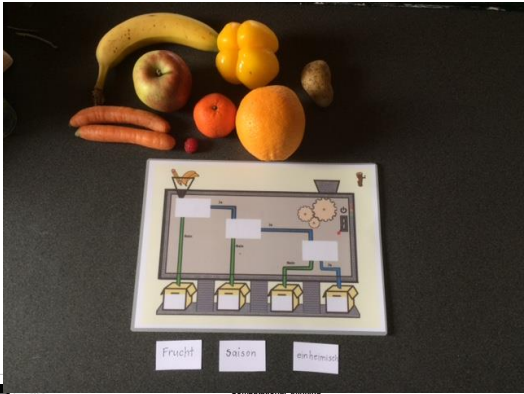
Computational thinking 16

Tri / Regroupement / Traitement de l'information

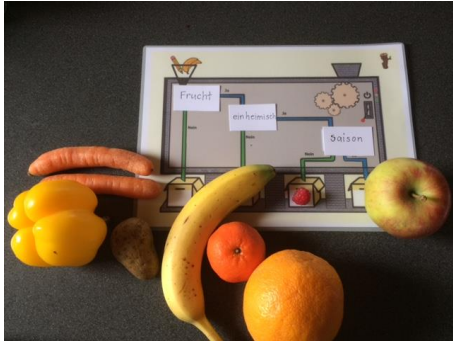
- Tri des objets selon des critères (collecte de données), récupération des objets - stratégies de tri - classifications
- Diviser les objets selon une procédure donnée
- Comparaison de deux objets
- Trouver des caractéristiques - établir des relations - appliquer des stratégies



Computational thinking 17



Computational thinking 18



Computational thinking 19

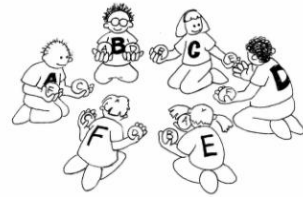


Computational thinking 20



Computational thinking 21

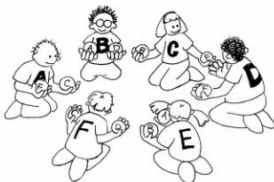
Routing / Deadlock



Computational thinking 22

Routing / Deadlock

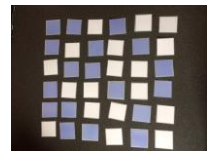
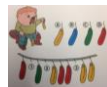
- transmission des données optimale / l'arrêt / bouchons de la transmission
- Les réseaux comme : Réseau téléphonique, réseau routier, systèmes informatiques
- Déverrouillage de l'arrêt / surcharge



Computational thinking 23

Régularités / Formes / Motifs

- La recherche de motifs dans certains objets est un outil important pour l'analyse de données, par exemple des textes, et est un métier de base en informatique.
- Les informaticiens sont des spécialistes pour résoudre les tâches d'optimisation.
- Idée de redondance / somme de contrôle : des informations supplémentaires sont là pour trouver d'éventuelles erreurs, mais ne font pas partie de l'information réelle.

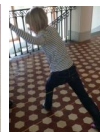


Computational thinking 24

Programmation



- Programmer signifie dire à une machine exactement ce qu'elle doit faire dans un langage de programmation.
- Les programmes informatiques sont les conséquences réelles des commandes. Les commandes sont exécutées exactement dans un ordre donné.
- Les programmeurs doivent être très précis et prudents pour atteindre l'objectif désiré. Une commande / séquence incorrecte peut déjà conduire à des résultats incorrects.
- Lire, comprendre et comprendre les programmes informatiques est l'une des compétences de base de l'informaticien.



Computational thinking

25



Computational thinking

26

Algorithme

Un algorithme est une séquence exacte d'instructions / commandes exécutables simples, étape par étape, pour résoudre un problème ou exécuter un ordre.

Les programmeurs conçoivent des algorithmes pour demander à l'ordinateur d'exécuter une tâche particulière.

Les algorithmes se retrouvent également dans les recettes de cuisine, les exercices de pliage, les opérations écrites - y compris le repassage des chemises.



Combinatoire

La combinatoire est une branche des mathématiques qui consiste à compter le nombre d'objets qui ont certaines propriétés.

Le but est de trouver des méthodes qui permettent de calculer facilement. Les informaticiens s'efforcent de les répertorier aussi systématiquement que possible.



Computational thinking

27



Computational thinking

28

Arbre de recherche - Sélection

Les arbres sont un outil important en informatique pour visualiser les données de manière structurée ou pour présenter clairement toutes les possibilités d'exécution d'une procédure.



Raisonnement logique

Le raisonnement logique est l'informatique pure (if ... then)

Pour les questions simples, facile à faire seule, pour de nombreuses hypothèses et conclusions, l'ordinateur est l'outil idéal pour garder une vue d'ensemble.

Les conséquences logiques sont particulièrement importantes lorsqu'il s'agit de la recherche de faute dans des programmes informatiques.



Computational thinking

29



Computational thinking

30