|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Zeit | Donnerstag | Freitag | WE | Montag | Dienstag | Mittwoch | Donnerstag | Freitag |
| 09.30 – 10.30 | **Java Basic** |  |  |  |  |  |  |  |
| 10.45 – 11.50 | **Planung** |  |  |  |  |  |  |  |
| 12.00 - 12.45 |  |  |  |  |  |  |  |  |
| Pause |  |  |  |  |  |  |  |  |
| 13.30 – 14.30 | **TU-Quizzen** |  |  | **TU-Quizzen** |  | **TU** | **TU** |  |
| 14.45 – 15.45 |  |  |  |  |
| 16.00 -17.00 |  |  |  |  |
| Zu Hause |  |  |  |  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Working with Data Types** | **Java Basic** | **Using Operators and Decision Constructs** | **Creating and Using Arrays** | **Using Loop Constructs** |
| **Working with Methods and Encapsulation** |  | **Handling Exceptions** | **Working with Selected classes from the Java API** |  |
| **Working with Inheritance** |  |  | **Quizfragen** | **Pause** |
| **Programmieren** |  |  |  |  |
|  |  |  |  |  |

**Aufgabe**

Erstelle deinen individuellen Lernplan. Achte darauf die Dinge auf jeden Fall zu wiederholen, welche dir besonders schwerfallen.

1. Fülle deine Selbsteinschätzung erneut aus. (Git) Und lade diese in Git auch wieder ausgefüllt hoch.
2. Mache dir eine Liste von Themen, welche du unbedingt wiederholen möchtest.
3. Fülle deinen Wochenplan mit den entsprechenden Oberthemen.

**Achtung**:

* + Du darfst **maximal drei** Blöcke als „Pause“ markieren.
  + Du solltest **mindestens drei** Blöcke Quizfragen beantworten.
  + Und **mindestens vier** verschiedene Themen wiederholen.
  + Priorisiere die Themen, welche du noch eher wenier behandelt hast.

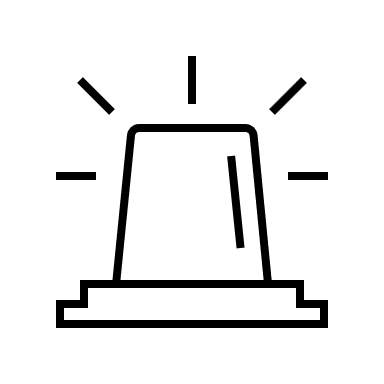
1. Lade deinen persönlichen Lernplan bei Git hoch! Ordner: \_20241216Bis20WeekPrep1811. Lernplan (In Unterricht)

**Was kannst du nutzen zum Lernen?**

* youTube
* <https://www.geeksforgeeks.org/>
* <https://www.w3schools.com/java/default.asp>
* <https://www.javatpoint.com/features-of-java>
* <https://openbook.rheinwerk-verlag.de/javainsel/>
* <https://wwwlehre.dhbw-stuttgart.de/~kfg/java/java.pdf>
* Bücher
* Alte Aufgaben und Folien
* Dozentin und Tutoren

**Verschiedene Aufgabensammlungen:**

* <https://www.thu.de/de/org/I/vorkurse/Documents/java_aufgaben.pdf>
* <https://tutego.de/javabuch/aufgaben/>
* <https://wiki.freitagsrunde.org/Javakurs/%C3%9Cbungsaufgaben>
* <https://info-wsf.de/uebungsaufgaben-erste-aufgaben-in-java/>
* <https://www.programmieraufgaben.ch/uploads/java.pdf>



**Wichtig**: Es muss jederzeit nachweisbar sein, was ihr gemacht habt!

|  |  |  |  |
| --- | --- | --- | --- |
| Topic | Detail | Wiederholen | Notizen |
| Java Basic |  |  |  |
|  | Define the scope of variables |  |  |
|  | Define the structure of a Java class |  |  |
|  | Create executable Java applications with a main method; run a Java program from the command line; produce console output |  |  |
|  | Import other Java packages to make them accessible in your code |  |  |
|  | Compare and contrast the features and components of Java such as: platform independence, object orientation, encapsulation, etc. |  |  |
| Working with Data Types | |  |  |
|  | Declare and initialize variables (including casting of primitive data types) |  |  |
|  | Differentiate between object reference variables and primitive variables |  |  |
|  | Know how to read or write to object fields |  |  |
|  | Explain an Object's Lifecycle (creation, "dereference by reassignment" and garbage collection) |  |  |
|  | Develop code that uses wrapper classes such as Boolean, Double, and Integer |  |  |
| Using Operators and Decision Constructs | |  |  |
|  | Use Java operators; use parentheses to override operator precedence |  |  |
|  | Test equality between Strings and other objects using == and equals () |  |  |
|  | Create if and if/else and ternary constructs |  |  |
|  | Use a switch statement |  |  |
| Creating and Using Arrays | |  |  |
|  | Declare, instantiate, initialize and use a one-dimensional array |  |  |
|  | Declare, instantiate, initialize and use multi-dimensional arrays |  |  |
| Using Loop Constructs | |  |  |
|  | Create and use while loops |  |  |
|  | Create and use for loops including the enhanced for loop |  |  |
|  | Create and use do/while loops |  |  |
|  | Compare loop constructs |  |  |
|  | Use break and continue |  |  |
| Working with Methods and Encapsulation | |  |  |
|  | Create methods with arguments and return values; including overloaded methods |  |  |
|  | Apply the static keyword to methods and fields |  |  |
|  | Create and overload constructors; differentiate between default and user defined constructors |  |  |
|  | Apply access modifiers |  |  |
|  | Apply encapsulation principles to a class |  |  |
|  | Determine the effect upon object references and primitive values when they are passed  into methods that change the values |  |  |
| Working with Inheritance | |  |  |
|  | Describe inheritance and its benefits |  |  |
|  | Develop code that makes use of polymorphism; develop code that overrides methods;  differentiate between the type of a reference and the type of an object |  |  |
|  | Determine when casting is necessary |  |  |
|  | Use super and this to access objects and constructors |  |  |
|  | Use abstract classes and interfaces |  |  |
| Handling Exceptions | |  |  |
|  | Differentiate among checked exceptions, unchecked exceptions, and Errors |  |  |
|  | Create a try-catch block and determine how exceptions alter normal program flow |  |  |
|  | Describe the advantages of Exception handling |  |  |
|  | Create and invoke a method that throws an exception |  |  |
|  | Recognize common exception classes (such as NullPointerException, ArithmeticException, ArrayIndexOutOfBoundsException, ClassCastException) |  |  |
| Working with Selected classes from the Java API | |  |  |
|  | Manipulate data using the StringBuilder class and its methods |  |  |
|  | Create and manipulate Strings |  |  |
|  | Create and manipulate calendar data using classes from java.time.LocalDateTime,  java.time.LocalDate, java.time.LocalTime, java.time.format.DateTimeFormatter, java.time.Period |  |  |
|  | Declare and use an ArrayList of a given type |  |  |
|  | Write a simple Lambda expression that consumes a Lambda Predicate expression |  |  |