

PORTFOLIO

Aufgabenstellung zum Kurs: Programmierung mit C/C++ (DLBROEPRS01_D)

INHALTSVERZEICHNIS

1. AUFGABENSTELLUNG.....	2
1.1. Aufgabe 1: Musik Ein HEX/BIN/CHAR-Datei-Editor	2
1.1.1. Konzeptionsphase.....	2
1.1.2. Erarbeitungs-/Reflexionsphase.....	3
1.1.3. Finalisierungsphase.....	3
1.2. Aufgabe 2: Mikrocontroller-Implementierung eines einfachen Kalkulators.....	4
1.2.1. Konzeptionsphase.....	4
1.2.2. Erarbeitungs-/Reflexionsphase.....	5
1.2.3. Finalisierungsphase.....	5
1.3. Aufgabe 3: Ein Viewer für Bode-Diagramme	6
1.3.1. Konzeptionsphase.....	6
1.3.2. Erarbeitungs-/Reflexionsphase.....	7
1.3.3. Finalisierungsphase.....	7
2. BETREUUNGSPROZESS	8
3. ZUSATZINFORMATIONEN ZUR BEWERTUNG	9
4. FORMALIA UND VORGABEN ZUR ABGABE	10
4.1. Bestandteile der Prüfungsleistung	10
4.2. Formalia zur Abgabe digitaler Dateien	10
4.3. Formalia für das Abstract.....	11

1. AUFGABENSTELLUNG

Es ist im Rahmen dieses Portfoliokurses eine der folgenden Aufgabenstellungen zu wählen.

Hinweis zum Urheberrecht und zur Plagiatsprüfung:

Es wird darauf hingewiesen, dass der IU Internationale Hochschule GmbH das Urheberrecht der Prüfungsaufgaben/Aufgabenstellungen obliegt. Einer Veröffentlichung der Aufgabenstellungen auf Drittplattformen wird ausdrücklich widersprochen. Im Falle einer Zuwiderhandlung stehen der Hochschule u.a. Unterlassungsansprüche zu. Zudem weisen wir darauf hin, dass jede eingereichte schriftliche Ausarbeitung mittels einer Plagiatssoftware überprüft wird. Wir empfehlen daher auch, keinesfalls ausgearbeitete Lösungen zu teilen, da dies den Verdacht eines Plagiates begründen kann.

1.1. Aufgabe 1: Musik Ein HEX/BIN/CHAR-Datei-Editor

Oft ist es sehr hilfreich, über einen Binärdateibetrachter zu verfügen, der es erlaubt, den Inhalt einer Datei zu visualisieren und ggf. die Möglichkeit der Bearbeitung zu ermöglichen. Daher soll in dieser Aufgabe ein Dateibetrachter/Editor entworfen und programmiert werden. Insbesondere soll der Dateiinhalt in hexadezimaler, binärer und zeichenweiser Form pro Dateispeicherort tabellarisch dargestellt werden und die Möglichkeit bestehen, die bearbeiteten Werte in den drei genannten Formen einzugeben. Neben der Möglichkeit, Änderungen zu speichern, kann die zu visualisierende Datei als Argument im Terminal übergeben oder als Parameter beim Start der Software eingegeben werden. Schließlich kann entweder C oder C++ als Programmiersprache gewählt werden und die Software kann als Terminal oder als Desktop-Anwendung realisiert werden.

Aufgabe: Entwurf und Realisierung eines Datei-Editors

Realisiere diese Aufgabe in den folgenden 3 Phasen.

1.1.1. Konzeptionsphase

Diese Phase ist der wichtigste Teil des Entwurfsprozesses. Alles, was in dieser Phase übersehen oder vergessen wird, wirkt sich später negativ auf die Umsetzung aus und führt im schlimmsten Fall zu unbrauchbaren Ergebnissen.

Der erste Schritt besteht darin, sich die Struktur der Software vorzustellen, von der Art der Interaktion zwischen der Software und den Benutzenden bis hin zu den grundlegenden Funktionalitäten, die bereitgestellt werden sollen. Dieser Schritt ist der Wichtigste, da die meisten Designentscheidungen an dieser Stelle getroffen werden. Es ist daher sehr wichtig, sich für diesen Schritt genügend Zeit zu nehmen und einzuplanen, bevor man mit den nächsten Schritten fortfährt.

Nachdem man sich die Interaktion zwischen Software und Benutzer:innen vorgestellt und die Hauptfunktionen der Software aufgelistet und beschrieben hat, ist es für die erfolgreiche Weiterführung des Projekts sehr hilfreich, die Hauptfunktionen der Software in Untermodule zu strukturieren. Eine ausführliche Beschreibung der einzelnen Module und der Interaktion zwischen ihnen dient als Leitfaden für den nächsten Schritt.

Nach der Vorstellung der Gesamtsoftware ist schließlich die Programmiersprache zu wählen, sei es C oder C++. Die Motivation für diese Wahl ist darzulegen.

Für die Einreichung ist ein konzeptioneller Text (1 DIN A4 Seite) zu erstellen, der diese Analysen und Überlegungen erläutert und das Zusammenspiel der Module und deren Interaktion detailliert beschreibt. Die PDF-Datei wird in das Abgabefeld der PebblePad Vorlage hochgeladen. Das Textfeld der Vorlage kann freigelassen werden.

Während des gesamten Prozesses werden Online-Tutorien angeboten, die die Möglichkeit bieten, sich auszutauschen, Ideen und/oder Entwürfe zu teilen und Feedback zu erhalten. In den Online-Tutorien können exemplarische Arbeiten mit den Tutor:innen besprochen werden. Hier haben alle Studierenden die Möglichkeit, sich einzubringen und von den Rückmeldungen der anderen zu lernen. Es wird empfohlen, diese Kanäle zu nutzen, um Fehler zu vermeiden und Verbesserungen vorzunehmen. Die Abgabe der Arbeit sollte erst erfolgen, nachdem Du die oben genannten Lern- und Informationsmedien genutzt hast. Danach erhältst Du eine Rückmeldung von den Tutor:innen und die Arbeit an der zweiten Phase kann beginnen.

1.1.2. Erarbeitungs-/Reflexionsphase

In dieser Phase beginnt die Programmierung der Software unter strikter Einhaltung des in der Konzeptionsphase gereiften Gesamtdesigns. Folgende Punkte sind zu verwirklichen:

- Jedes geplante Modul wird programmiert und getestet
- Interaktionen zwischen den Modulen werden programmiert und getestet
- Die Gesamtstruktur der Software und die Benutzerinteraktion wird realisiert und getestet
- Der Code muss ordentlich kommentiert sein
- Der Code muss lesbar sein, d.h. Verwendung geeigneter Variablennamen, Beibehaltung eines bestehenden Codelayouts, korrekte Verwendung von Klammern und Zeilenumbrüchen

Als Ergebnis dieser Phase ist die erste Version der Software fertig. In diesem Stadium sollte die Software betriebsbereit sein und korrekt funktionieren. Wenn ein Fehler festgestellt wird, sollte er behoben werden, bevor der Abschluss dieser Phase eingereicht wird.

Der vollständige Code wird in einen GIT-Dienst (GitHub, GitLab) hochgeladen und ein PDF-Dokument, das die Struktur der Software und die wichtigsten Anmerkungen beschreibt, muss bereitgestellt werden. Der Link zum GIT-Projekt muss in das PDF-Dokument aufgenommen werden. Falls erforderlich, sind auch Anweisungen für die Kompilierung, die Ausführung und den Test der Software zu beschreiben. Das PDF-Dokument wird in das Abgabefeld der PebblePad Vorlage hochgeladen. Das Textfeld in der Vorlage kann freigelassen werden.

Während des gesamten Prozesses bieten Online-Tutorien und andere Kanäle die Möglichkeit, Ideen und/oder Entwürfe eingehend zu diskutieren und ausreichend Feedback, Tipps und Hinweise zu erhalten. Es wird empfohlen, diese Kanäle zu nutzen, um Fehler zu vermeiden und Deine Arbeit zu verbessern. Ist dies geschehen, kannst Du deine zweite Phase zur Bewertung einreichen. Nach einer Rückmeldung der Tutor:innen wird die Arbeit am endgültigen Entwurf in der dritten Phase fortgesetzt.

1.1.3. Finalisierungsphase

In der Finalisierungsphase geht es darum, den Dateieditor zu optimieren und mögliche Fehler zu beheben, nachdem die Tutor:innen ein Feedback gegeben haben, und die Aufgabe abzuschließen. Bestimmte Elemente müssen eventuell noch einmal verbessert oder geändert werden.

Zusätzlich ist eine Zusammenfassung erwünscht, in der die wichtigsten Entscheidungen und Lösungen, die für die Realisierung der Aufgabe getroffen wurden, beschrieben werden. Der vollständige Code wird weiterhin in denselben GIT-Dienst hochgeladen, der auch in der vorherigen Phase verwendet wurde. Alle erarbeiteten Ergebnisse werden gemeinsam mit dem Link zum Git-Dienst in eine PDF-Datei hochgeladen und in das Abgabefeld des finalen Produkts hochgeladen. Zusätzlich ist ein Abstract gewünscht, welches das abgeschlossene Projekt und alle seine Teile beschreibt. Das Einfügen eines zip-Ordners ist nicht notwendig.

In der Finalisierungsphase bieten die Online-Tutorien und andere Kanäle ebenfalls die Möglichkeit, ausreichend Feedback, Tipps und Hinweise zu erhalten, bevor das fertige Produkt endgültig abgegeben wird. Es empfiehlt sich, diese Kanäle zu nutzen, um Fehler zu vermeiden und Verbesserungen vorzunehmen. Das fertige Produkt wird zusammen mit den Ergebnissen aus Phase 1 und Phase 2 und den oben genannten Materialien eingereicht. Nach der Einreichung der dritten Portfolioseite geben die Tutor:innen innerhalb von sechs Wochen eine abschließende Rückmeldung, die eine Bewertung und Benotung enthält.

1.2. Aufgabe 2: Mikrocontroller-Implementierung eines einfachen Kalkulators

In dieser Aufgabe geht es darum, einen Mikrocontroller für die Durchführung grundlegender algebraischer Berechnungen mit zwei Termen, d.h. $34 * 72 = 2448$, mittels einer seriellen Schnittstelle (RS232, UART) zwischen einem Personal Computer (PC) und einer Mikrocontroller-Platine (uC) einzusetzen. Daher beinhaltet diese Aufgabe den Entwurf und die Realisierung von zwei verschiedenen Anwendungen, eine für den PC und eine für den uC. Nach dem Aufbau einer seriellen Kommunikation mit dem uC ermöglicht die PC-Anwendung das Senden von Strings, die die auszuführende Berechnung enthalten, wie z. B. " $34 * 72$ ". An diesem Punkt parst das uC den Ausdruck und berechnet, falls gültig, das Ergebnis und sendet es zur Visualisierung an den PC zurück. Der gesamte Nachrichtenaustausch zwischen PC und uC kann dann in einer Textdatei gespeichert werden. Die uC-Anwendung ist in C zu programmieren, während die PC-Anwendung entweder in C oder in C++ programmiert werden kann. Als uC-Boards eignen sich Arduino- oder ST Nucleo-Boards, sofern eine serielle Peripherie und eine Kommunikationsschnittstelle vorhanden sind.

Aufgabe: Entwurf und Realisierung eines einfachen Mikrocontroller-Rechners

Realisiere diese Aufgabe in den folgenden 3 Phasen.

1.2.1. Konzeptionsphase

Diese Phase ist der wichtigste Teil des Entwurfsprozesses. Alles, was in dieser Phase übersehen oder vergessen wird, wirkt sich später negativ auf die Umsetzung aus und führt im schlimmsten Fall zu unbrauchbaren Ergebnissen.

Der erste Schritt besteht darin, sich die Struktur beider Software vorzustellen, von der Art der Interaktion zwischen der Software und den Benutzer:innen bis hin zu den grundlegenden Funktionalitäten, die bereitgestellt werden sollen. Dieser Schritt ist der wichtigste, da hier die meisten Designentscheidungen getroffen werden. Es ist daher sehr wichtig, sich für diesen Schritt genügend Zeit zu nehmen, bevor mit den nächsten Schritten fortgegangen wird.

Nachdem man sich die Interaktion zwischen Software und Benutzer:innen vorgestellt und die Hauptfunktionen der Software aufgelistet und beschrieben hat, ist es für die erfolgreiche Weiterführung des Projekts sehr hilfreich, die Hauptfunktionen der Software in Untermodule für beide Anwendungen zu strukturieren. Eine ausführliche Beschreibung der einzelnen Module und ihrer Interaktion untereinander dient als Leitfaden für den nächsten Schritt.

Nach der Vorstellung der Gesamtsoftware muss schließlich die Programmiersprache für die PC-Anwendung gewählt werden, d.h. C oder C++. Diese Wahl ist zu begründen.

Für die Einreichung ist ein konzeptioneller Text (1 DIN A4 Seite) zu erstellen, der diese Analysen und Überlegungen erläutert und das Zusammenspiel der Module und deren Interaktion detailliert beschreibt. Die PDF-Datei wird in das Abgabefeld der PebblePad Vorlage hochgeladen. Das Textfeld der Vorlage kann freigelassen werden.

Während des gesamten Prozesses werden Online-Tutorien angeboten, die die Möglichkeit bieten, zu diskutieren, Ideen und/oder Entwürfe auszutauschen und Feedback zu erhalten. In den Online-Tutorien können exemplarische Arbeiten mit den Tutor:innen besprochen werden. Hier haben alle Studierenden die Möglichkeit, sich einzubringen und von den Rückmeldungen der anderen zu lernen. Es wird empfohlen, diese Kanäle zu nutzen, um Fehler zu vermeiden und Verbesserungen vorzunehmen. Die Abgabe der Arbeit sollte erst erfolgen, nachdem Du die oben genannten Lern- und Informationsmedien genutzt hast. Danach erhältst Du eine Rückmeldung von den Tutor:innen und die Arbeit an der zweiten Phase kann beginnen.

1.2.2. Erarbeitungs-/Reflexionsphase

In dieser Phase beginnt die Programmierung der Software unter strikter Einhaltung des in der Konzeptionsphase gereiften Gesamtdesigns. Folgende Punkte sind zu verwirklichen:

- Jedes geplante Modul wird programmiert und getestet
- Interaktionen zwischen den Modulen werden programmiert und getestet
- Die Gesamtstruktur der Software und die Benutzerinteraktion wird realisiert und getestet
- Der Code muss ordentlich kommentiert sein
- Der Code muss lesbar sein, d.h. Verwendung geeigneter Variablennamen, Beibehaltung eines bestehenden Codelayouts, korrekte Verwendung von Klammern und Zeilenumbrüchen

Als Ergebnis dieser Phase sind die ersten Versionen der Software fertig. In diesem Stadium sollte die Software betriebsbereit sein und korrekt funktionieren. Wenn ein Fehler festgestellt wird, sollte er behoben werden, bevor der Abschluss dieser Phase eingereicht wird.

Der vollständige Code wird in einen GIT-Dienst (GitHub, GitLab) hochgeladen und ein PDF-Dokument, das die Struktur der Software und die wichtigsten Anmerkungen beschreibt, muss bereitgestellt werden. Der Link zum GIT-Projekt muss in das PDF-Dokument aufgenommen werden. Falls erforderlich, sind auch Anweisungen für die Kompilierung, die Ausführung und den Test der Software zu beschreiben. Das PDF-Dokument wird in das Abgabefeld der PebblePad Vorlage hochgeladen. Das Textfeld in der Vorlage kann freigelassen werden.

Während des gesamten Prozesses bieten Online-Tutorien und andere Kanäle die Möglichkeit, Ideen und/oder Entwürfe eingehend zu diskutieren und ausreichend Feedback, Tipps und Hinweise zu erhalten. Es wird empfohlen, diese Kanäle zu nutzen, um Fehler zu vermeiden und Deine Arbeit zu verbessern. Ist dies geschehen, kannst Du Deine zweite Phase zur Bewertung einreichen. Nach einer Rückmeldung der Tutor:innen wird die Arbeit am endgültigen Entwurf in der dritten Phase fortgesetzt.

1.2.3. Finalisierungsphase

In der Finalisierungsphase geht es darum, das Gesamtsystem zu optimieren und eventuelle Fehler zu beheben, nachdem man erneut Feedback von den Tutor:innen erhalten hat, und die Aufgabe abzuschließen. Bestimmte Elemente müssen eventuell noch einmal verbessert oder geändert werden.

Zusätzlich ist eine Zusammenfassung erwünscht, in der die wichtigsten Entscheidungen und Lösungen, die für die Realisierung der Aufgabe getroffen wurden, beschrieben werden. Der vollständige Code wird weiterhin in denselben GIT-Dienst hochgeladen, der auch in der vorherigen Phase verwendet wurde. Alle erarbeiteten Ergebnisse werden gemeinsam mit dem Link zum Git-Dienst in eine PDF-Datei hochgeladen und in das Abgabefeld des finalen Produkts hochgeladen. Zusätzlich ist ein Abstract gewünscht, welches das abgeschlossene Projekt und alle seine Teile beschreibt. Das Einfügen eines zip-Ordners nicht notwendig.

In der Finalisierungsphase bieten die Online-Tutorien und andere Kanäle ebenfalls die Möglichkeit, ausreichend Feedback, Tipps und Hinweise zu erhalten, bevor das fertige Produkt endgültig abgegeben wird. Es empfiehlt sich,

diese Kanäle zu nutzen, um Fehler zu vermeiden und Verbesserungen vorzunehmen. Das fertige Produkt wird zusammen mit den Ergebnissen aus Phase 1 und Phase 2 und den oben genannten Materialien eingereicht. Nach der Einreichung der dritten Portfolioseite geben die Tutor:innen innerhalb von sechs Wochen eine abschließende Rückmeldung, die eine Bewertung und Benotung enthält.

1.3. Aufgabe 3: Ein Viewer für Bode-Diagramme

In dieser Aufgabe soll eine Software entworfen und realisiert werden, die in der Lage ist, das Bode-Diagramm einer generischen Übertragungsfunktion darzustellen. Die Software berechnet nach der Einführung einer gegebenen Übertragungsfunktion numerisch das zugehörige Bode-Diagramm, d.h. Amplituden- und Phasendiagramme, innerhalb eines bestimmten Frequenzbereichs und stellt es dar. Darüber hinaus sollen Amplituden- und Phasenstabilitätsspannen berechnet werden, um die Stabilität der Übertragungsfunktion zu bewerten. Von der Software wird auch erwartet, dass sie die angezeigten Diagramme in einem Bildformat, wie JPG oder PNG, oder in einem vektoriellen Grafikformat, wie SVG oder PDF, exportieren kann.

In diesem Fall ist es wünschenswert, eine C++-basierte Desktop-Programmierungsumgebung (z. B. Qt) und Bibliotheken von Drittanbietern für die Handhabung von Bildformaten zu verwenden.

Aufgabe: Entwurf und Realisierung eines Bode-Diagramm-Rechners

Realisiere diese Aufgabe in den folgenden 3 Phasen.

1.3.1. Konzeptionsphase

Diese Phase ist der wichtigste Teil des Entwurfsprozesses. Alles, was in dieser Phase übersehen oder vergessen wird, wirkt sich später negativ auf die Umsetzung aus und führt im schlimmsten Fall zu unbrauchbaren Ergebnissen.

Der erste Schritt besteht darin, sich die Struktur der Software vorzustellen, von der Art der Interaktion zwischen der Software und den Benutzenden bis hin zu den grundlegenden Funktionalitäten, die bereitgestellt werden sollen. Dieser Schritt ist der wichtigste, da die meisten Designentscheidungen an dieser Stelle getroffen werden. Es ist daher sehr wichtig, sich für diesen Schritt genügend Zeit zu nehmen, bevor man mit den nächsten Schritten fortfährt.

Nachdem man sich die Interaktion zwischen Software und Benutzer:innen vorgestellt und die Hauptfunktionen der Software aufgelistet und beschrieben hat, ist es für die erfolgreiche Weiterführung des Projekts sehr hilfreich, die Hauptfunktionen der Software in Untermodule zu strukturieren. Eine ausführliche Beschreibung der einzelnen Module und der Interaktion zwischen ihnen dient als Leitfaden für den nächsten Schritt.

Nach der Vorstellung der Gesamtsoftware muss schließlich die Desktop-Programmierungsumgebung ausgewählt werden (die Qt-Framework-Umgebung wird sehr empfohlen). Die Motivation für diese Wahl ist darzulegen.

Für die Einreichung ist ein konzeptioneller Text (1 DIN A4 Seite) zu erstellen, der diese Analysen und Überlegungen erläutert und das Zusammenspiel der Module und deren Interaktion detailliert beschreibt. Die PDF-Datei wird in das Abgabefeld der PebblePad Vorlage hochgeladen. Das Textfeld der Vorlage kann freigelassen werden.

Während des gesamten Prozesses werden Online-Tutorien angeboten, die die Möglichkeit bieten, zu diskutieren, Ideen und/oder Entwürfe auszutauschen und Feedback zu erhalten. In den Online-Tutorien können exemplarische Arbeiten mit den Tutor:innen besprochen werden. Hier haben alle Studierenden die Möglichkeit, sich einzubringen und von den Rückmeldungen der anderen zu lernen. Es wird empfohlen, diese Kanäle zu nutzen, um Fehler zu vermeiden und Verbesserungen vorzunehmen. Die Abgabe der Arbeit sollte erst erfolgen, nachdem Du die

oben genannten Lern- und Informationsmedien genutzt hast. Danach erhältst Du eine Rückmeldung von den Tutor:innen und die Arbeit an der zweiten Phase kann beginnen.

1.3.2. Erarbeitungs-/Reflexionsphase

In dieser Phase beginnt die Programmierung der Software unter strikter Einhaltung des in der Konzeptionsphase gereiften Gesamtdesigns. Folgende Punkte sind zu verwirklichen:

- Jedes geplante Modul wird programmiert und getestet
- Interaktionen zwischen den Modulen werden programmiert und getestet
- Die Gesamtstruktur der Software und die Benutzerinteraktion wird realisiert und getestet
- Der Code muss ordentlich kommentiert sein
- Der Code muss lesbar sein, d.h. Verwendung geeigneter Variablennamen, Beibehaltung eines bestehenden Codelayouts, korrekte Verwendung von Klammern und Zeilenumbrüchen

Als Ergebnis dieser Phase ist die erste Version der Software fertig. In diesem Stadium sollte die Software betriebsbereit sein und korrekt funktionieren. Wenn ein Fehler festgestellt wird, sollte er behoben werden, bevor der Abschluss dieser Phase eingereicht wird.

Der vollständige Code wird in einen GIT-Dienst (GitHub, GitLab) hochgeladen und ein PDF-Dokument, das die Struktur der Software und die wichtigsten Anmerkungen beschreibt, muss bereitgestellt werden. Der Link zum GIT-Projekt muss in das PDF-Dokument aufgenommen werden. Falls erforderlich, sind auch Anweisungen für die Kompilierung, die Ausführung und den Test der Software zu beschreiben. Das PDF-Dokument wird in das Abgabefeld der PebblePad Vorlage hochgeladen. Das Textfeld in der Vorlage kann freigelassen werden.

Während des gesamten Prozesses bieten Online-Tutorien und andere Kanäle die Möglichkeit, Ideen und/oder Entwürfe eingehend zu diskutieren und ausreichend Feedback, Tipps und Hinweise zu erhalten. Es wird empfohlen, diese Kanäle zu nutzen, um Fehler zu vermeiden und Deine Arbeit zu verbessern. Ist dies geschehen, kannst Du Deine zweite Phase zur Bewertung einreichen. Nach einer Rückmeldung der Tutor:innen wird die Arbeit am endgültigen Entwurf in der dritten Phase fortgesetzt.

1.3.3. Finalisierungsphase

In der Finalisierungsphase geht es darum, das Gesamtsystem zu optimieren und eventuelle Fehler zu beheben, nachdem man erneut Feedback von den Tutor:innen erhalten hat, und die Aufgabe abzuschließen. Bestimmte Elemente müssen eventuell noch einmal verbessert oder geändert werden.

Zusätzlich ist eine Zusammenfassung erwünscht, in der die wichtigsten Entscheidungen und Lösungen, die für die Realisierung der Aufgabe getroffen wurden, beschrieben werden. Der vollständige Code wird weiterhin in denselben GIT-Dienst hochgeladen, der auch in der vorherigen Phase verwendet wurde. Alle erarbeiteten Ergebnisse werden gemeinsam mit dem Link zum Git-Dienst in eine PDF-Datei hochgeladen und in das Abgabefeld des finalen Produkts hochgeladen. Zusätzlich ist ein Abstract gewünscht, welches das abgeschlossene Projekt und alle seine Teile beschreibt. Das Einfügen eines zip-Ordners ist nicht notwendig.

In der Finalisierungsphase bieten die Online-Tutorien und andere Kanäle ebenfalls die Möglichkeit, ausreichend Feedback, Tipps und Hinweise zu erhalten, bevor das fertige Produkt endgültig abgegeben wird. Es empfiehlt sich, diese Kanäle zu nutzen, um Fehler zu vermeiden und Verbesserungen vorzunehmen. Das fertige Produkt wird zusammen mit den Ergebnissen aus Phase 1 und Phase 2 und den oben genannten Materialien eingereicht. Nach der Einreichung der dritten Portfolioseite geben die Tutor:innen innerhalb von sechs Wochen eine abschließende Rückmeldung, die eine Bewertung und Benotung enthält.

2. BETREUUNGSPROZESS

Bei der Betreuung der Portfolios stehen grundsätzlich mehrere Kanäle offen. Die jeweilige Inanspruchnahme liegt dabei im eigenen Verantwortungsbereich. Die eigenständige Erarbeitung eines Produktes und die Befüllung der jeweiligen Portfolioteile ist dabei Teil der zu erbringenden Prüfungsleistung und fließt in die Gesamtbewertung mit ein.

Zum einen sieht die tutorielle Betreuung Feedbackschleifen zu den einzureichenden Portfolioteilen im Rahmen der Konzeptions- sowie der Erarbeitungs- und Reflexionsphase vor. Das Feedback erfolgt im Rahmen einer Einreichung des jeweiligen Portfolioteils. Des Weiteren werden regelmäßige Online-Tutorien angeboten, in denen Gelegenheit besteht, mit den Tutor:innen Fragen zur Bearbeitung des Portfolios zu besprechen. Die Tutor:innen stehen zusätzlich für fachliche Rücksprachen sowie für formale und allgemeine Fragen zum Vorgehen bei der Portfoliobearbeitung zur Verfügung.

Technische Fragen zur Nutzung von PebblePad sind per Mail an das Prüfungsamt zu richten.

3. ZUSATZINFORMATIONEN ZUR BEWERTUNG

In die Bewertung des Portfolios fließen die folgenden Kriterien mit dem jeweils angegebenen Prozentsatz ein:

Bewertungskriterien	Erläuterungen	Gewichtung
Problemabgrenzung/Zielsetzung	*Erfassung des Problems *Klare Problemabgrenzung/Zielsetzung *Nachvollziehbares Konzept	10%
Methodik/Idee/Vorgehen	*Angemessener Transfer von Theorien/Modellen *Klare Angaben zur gewählten Methodik/zur gewählten Idee/zum gewählten Vorgehen	20%
Qualität der Umsetzung	*Qualität der Umsetzung und Dokumentation	40%
Kreativität/Richtigkeit	*Kreativität des Lösungsansatzes *Umgesetzte Lösung erfüllt angestrebte Zielsetzung	20%
Formale Anforderungen	*Einhaltung der formalen Vorgaben.	10%

Bei der Konzeption und Erstellung des Portfolios sollten die genannten Bewertungskriterien einschließlich der folgenden Erläuterungen berücksichtigt werden.

Problemabgrenzung/Zielsetzung: Fähigkeit, die wichtigsten Herausforderungen der Aufgabe zu verstehen und eine Strategie und Lösung zu entwerfen, die als Grundlage für die Entwicklung der Aufgabe dient.

Methodik/Idee/Vorgehen: Fähigkeit zur Umsetzung der geplanten Entwicklungsstrategie in eine konkrete Softwarestruktur mit genau definierten Modulen und deren Interaktion.

Qualität der Umsetzung: Trotz der Funktionalität ist es sehr wichtig, lesbaren und eleganten Code zu erstellen, der von anderen leicht verstanden und geändert werden kann. Darüber hinaus spielt die Wahl der Programmierung, z. B. der implementierten Algorithmen und Werkzeuge, eine wichtige Rolle bei der Bewertung der Qualität der Implementierung.

Kreativität/Richtigkeit: Fähigkeit, kreative Lösungen für die identifizierten Herausforderungen zu finden, die entweder die Entwicklung der Aufgabe vereinfachen oder die Möglichkeit bieten, weitere Funktionalitäten, die nicht in der Aufgabenbeschreibung aufgeführt sind, einfacher zu implementieren.

Formale Anforderungen: Das Folgende ist sowohl in den Code-Projekten (mit Verweis auf Code-Kommentare) als auch im erstellten Dokument zu beachten. Die Grammatik und die Syntax sollen fehlerfrei sein. Die Beschreibung des Inhalts muss gut strukturiert und für andere stets leicht lesbar sein. Lange verschachtelte Sätze sind nicht zu empfehlen.

4. FORMALIA UND VORGABEN ZUR ABGABE

4.1. Bestandteile der Prüfungsleistung

Im Folgenden befindet sich eine Übersicht der Prüfungsleistung Portfolio mit seinen einzelnen Phasen, einzureichenden Einzelleistungen und Feedbackrunden im Überblick. Für die Erarbeitung der Portfolioteile im Rahmen der Prüfungsleistung wird eine Vorlage in PebblePad zur Verfügung gestellt. Die Vorlage ist Bestandteil dieser Prüfungsleistung.

Phase	Zwischenergebnis	Einzureichende Leistung
Konzeptionsphase	Portfolioteil 1	<ul style="list-style-type: none"> Hochladen einer 1-seitigen Konzeptpräsentation (Datei im pdf-Format) in schriftlicher Form, in der die Hauptmodule des Konzepts beschrieben werden
		Feedback
Erarbeitungsphase/ Reflexionsphase	Portfolioteil 2	<ul style="list-style-type: none"> Erläuterung der Umsetzung in schriftlicher Form (keine feste Längenbegrenzung) als PDF-Datei
		Feedback
Finalisierungsphase	Portfolioteil 3	<ul style="list-style-type: none"> Eine 2-seitige Zusammenfassung (Abstract), die das abgeschlossene Projekt und alle seine Teile beschreibt Finales Produkt (Maßnahme) als PDF Ergebnis aus Phase 1 Ergebnis aus Phase 2
		Feedback + Note

4.2. Formalia zur Abgabe digitaler Dateien

Konzeptionsphase

Empfohlene Hilfsmittel/Software zur Bearbeitung

Zugelassene Dateiformate

Die Dokumentation wird als PDF- oder Word-Dokument hochgeladen.

Dateigröße

Nicht begrenzt

Weitere Formalien und Parameter

Dateien sind immer nach folgendem Muster zu benennen:

Für die prüfungsleistungsrelevanten Abgaben auf PebblePad:

Nachname-Vorname_Matrikelnummer_Kurs_P(hase)-1_A(bgabe)

Beispiel: Mustermann-Max_12345678_Programmierung mit C/C++ _P1_A

Erarbeitungs-/Reflexionsphase

Empfohlene Hilfsmittel/Software zur Bearbeitung	Software: Die Programmierumgebung kann in Absprache mit den Tutor:innen gewählt werden. Empfohlene Plattformen sind jedoch das Qt-Framework und der QtCreator sowie Microsoft Visual Studio Hardware: Arduino- oder ST Nucleo-Platinen
---	---

Zugelassene Dateiformate	Die Dokumentation wird als PDF- oder Word-Dokument hochgeladen.
--------------------------	---

Dateigröße	Nicht begrenzt
------------	----------------

Weitere Formalien und Parameter	Dateien sind immer nach folgendem Muster zu benennen:
---------------------------------	---

Für die prüfungsleistungsrelevanten Abgaben auf PebblePad:

Nachname-Vorname_Matrikelnummer_Kurs_P(hase)-2_A(bgabe)

Beispiel: Mustermann-Max_12345678_Programmierung mit C/C++ _P2_A

Finalisierungsphase

Empfohlene Hilfsmittel/Software zur Bearbeitung	Software: Die Programmierumgebung kann in Absprache mit den Tutor:innen gewählt werden. Empfohlene Plattformen sind jedoch das Qt-Framework und der QtCreator sowie Microsoft Visual Studio Hardware: Arduino- oder ST Nucleo-Platinen
---	---

Zugelassene Dateiformate	Die Dokumentation wird als PDF- oder Word-Dokumente hochgeladen.
--------------------------	--

Dateigröße	Nicht begrenzt
------------	----------------

Weitere Formalien und Parameter	Dateien sind immer nach folgendem Muster zu benennen:
---------------------------------	---

Für die prüfungsleistungsrelevanten Abgaben auf PebblePad:

Nachname-Vorname_Matrikelnummer_Kurs_P(hase)-3_A(bgabe)

Beispiel: Mustermann-Max_12345678_Programmierung mit C/C++ _P3_A

4.3. Formalia für das Abstract

Umfang	2 Seiten Textteil
Papierformat	DIN A4
Seitenränder	Oben und unten 2cm; links 2cm; rechts 2cm
Schrifttyp	Allgemeiner Text – Arial 11Pkt; Überschriften – 12Pkt, Blocksatz
Zeilenabstand	1,5
Satz	Blocksatz und Silbentrennung
Fußnoten	Arial 10Pkt, Blocksatz
Absätze	Nach gedanklicher Gliederung – 6Pkt Abstand nach Zeilenumbruch
Eidesstattliche Erklärung	Die Abgabe der Eidesstattlichen Erklärung erfolgt in elektronischer Form über myCampus. Davor ist keine Einreichung der Prüfungsleistung möglich.

Bitte beachtet hierzu die Anleitung für das Einreichen eines Portfolios in myCampus.

Bei Fragen zur Abgabe des Portfolios wende Dich bitte per Mail an das Prüfungsamt.

Beachte bitte zusätzlich die Nutzungsanleitung zu PebblePad & Atlas!

Viel Erfolg beim Erstellen des Portfolios!