

プログラミング言語論レポート

2025年8月7日

学籍番号：35714121

名前：福富隆大

レポート4-1：クラスを使ってみる

課題内容

名前と学籍番号をデータとして持ち、`show`メソッドで名前と学籍番号を表示できるクラスを作成し、`main`関数でその機能を確認する。

実装解説

`Student`クラスを作成し、C++の`std::string`を使用して名前と学籍番号を管理しました。`private`メンバ変数として名前と学籍番号を持ち、`public`メソッドとして`setName`、`setStudentNumber`、`show`を実装しました。

実装ファイル

- **report4-1.cpp:** `Student`クラスの実装

レポート4-2：分割コンパイル

課題内容

スタッククラスを3つのファイル（ヘッダファイル、実装ファイル、メインプログラム）に分割し、`Makefile`を作成して分割コンパイルを行う。また、要素数を返すメソッドを追加する。

実装解説

スタッククラスをヘッダファイル（`stack.hpp`）と実装ファイル（`stack.cpp`）に分割し、メインプログラム（`main.cpp`）で使用しました。要素数を返す`size()`メソッドを追加し、`Makefile`でコンパイルできるようにしました。

実装ファイル

- **stack.hpp:** スタッククラスの宣言
- **stack.cpp:** スタッククラスの実装
- **main.cpp:** テストプログラム
- **Makefile:** コンパイル設定ファイル

レポート5-1：コンストラクタ・デストラクタ

課題内容

教室の設備（席、机、プロジェクタ、黒板の数）をメンバ変数を持つクラスを作成し、デフォルトコンストラクタとパラメータ付きコンストラクタを実装する。

実装解説

Kyoshitsuクラスを作成し、コンストラクタのオーバーロードを実装しました。デフォルトコンストラクタではすべて0で初期化し、パラメータ付きコンストラクタでは指定された値で初期化します。デストラクタでメッセージを表示して確認できるようにしました。

実装ファイル

- **report5-1.cpp**: Kyoshitsuクラスの実装

レポート5-2：継承

課題内容

与えられたIntArrayクラスを継承し、配列要素に値を設定するsetメソッドと合計を計算するsumメソッドを持つ新しいクラスを作成する。

実装解説

IntArrayクラスのprivateメンバをprotectedに変更し、ExtendedIntArrayクラスで継承しました。set()メソッドで配列の要素に値を設定し、sum()メソッドで全要素の合計を計算します。境界チェックも含めました。

実装ファイル

- **report5-2.cpp**: ExtendedIntArrayクラスの継承実装

レポート6-1：抽象クラスと多態性

課題内容

純粋仮想関数area()を持つ抽象クラスObjectを定義し、RectangleクラスとCircleクラスを派生クラスとして実装する。

実装解説

抽象クラスObjectに純粋仮想関数area()を定義し、RectangleクラスとCircleクラスでそれぞれ実装しました。Rectangleクラスは対角2点の座標から面積を計算し、Circleクラスは半径から面積を計算します。多態性を利用したテストも含めました。

実装ファイル

- **report6-1.cpp**: 抽象クラスObjectとRectangle・Circleクラスの実装

レポート6-2：ジェネリックな関数

課題内容

与えられたComparableインターフェースとDoubleクラスを使用し、find_min関数を改良して最小値を表示する機能を実装する。さらにCharクラスとPersonクラスを追加実装する。

実装解説

提供されたmin1.cppを基に、以下の5つの課題を実装しました：

1. 元のコードの動作確認
2. find_min()関数の改良（最小値表示機能追加）
3. Charクラスの実装
4. Charクラス配列でのfind_min動作確認
5. Personクラス（年齢最小）の実装

各クラスはComparableインターフェースのLTE()メソッド（Less Than or Equal）とprint()メソッドを実装し、同一のfind_min()関数で動作するように設計しました。

実装ファイル

- **report6-2.cpp:** 全ての課題を含む完全な実装

まとめ

本レポートでは、C++のオブジェクト指向プログラミングの基本概念を実装を通じて学習しました：

1. **レポート4:** クラスの基本構造と分割コンパイルを学習
2. **レポート5:** コンストラクタ・デストラクタと継承の仕組みを理解
3. **レポート6:** 抽象クラス、多態性、ジェネリックプログラミングを実践

各課題を通じて、C++における重要な概念であるカプセル化、継承、多態性を実装し、オブジェクト指向設計の理解を深めることができました。