

プログラミング グループ課題

2024/12/12 遠山

追記：2025/12/08 野村

目次

プログラミング グループ課題.....	1
全体概要.....	2
ひな型の概要.....	3
本課題で学習する開発の流れ	5
仕様（策定）・設計・実装・テスト	5
設計・実装・テストの例.....	6
本課題の必須機能の仕様.....	8
必須機能 1：BGM と SE の設定	8
必須機能 2：キャラクターの置き換え&アイテム作り	8
必須機能 3：新マップの生成	8
必須機能 4：アイテムの配置	8
必須機能 5：ゴールの作成	8
ひな型の実行方法	9
今後のスケジュール	10
1.準備（12/12）	10
2.構想発表	11
3.必須機能のみのゲームの提出	11
4.独自機能の設計・テストの提出	11
5.最終発表会	11
6.発表会後	11
注意事項.....	11
今後の提出物.....	11
個人レポート（最終発表会後）	12
グループレポート（最終発表会後、分担して執筆したものを代表者 1 名が提出）	12

全体概要

この資料ではグループ課題について説明します。みなさんには 5 名程度のグループで課題に取り組んでもらい、その成果を構想発表や最終発表会で披露してもらいます。ベースとなるプログラムは提供しますが、みんなの力でそれを発展させ、他のグループとは異なるできるだけ面白いゲームを作ってください。グループメンバーは教員が決めました。メンバー表は授業資料の中に入っていますので確認してください。

本課題は、全員がきちんと参加するという前提において、1名で進めるよりも複数名で分担し意思疎通しながら進めた方が完成度の高いものができるようにしています。得意・不得意はあると思いますが、不得意だという自覚がある人ほど、自分にできることを探してグループに貢献できるようにしましょう。力を出し合って質の高い作品を完成させることを目指してください。

本課題で最も難しいと考えられるのは、それぞれの作業でできたものを組み合わせたり、組み合わせの際のバグを取り除いたりといった結合の部分です。余裕をもって結合にのぞむことができるよう、個人の分担箇所で困った場合には可能な限り早く相談するようにしてください。

課題には「与えられた仕様に沿って必須機能を実装する」部分と「独自機能の仕様を策定して実装する」部分とがあります。**5つの必須機能を実装すること、2つ以上の独自機能を考えて実装することが条件です。** 必須機能が適切に実現されていること、興味深い独自機能が多く実現されていること、発表会でのトーク・デモ・質疑応答がうまくできていることが、それぞれ高評価を得るためのポイントです。必要に応じて [Github¹](#) や、JDB (Java のデバッガ)、Scene Builder (JavaFX の GUI を構築するための GUI 環境)²など使用するのは大歓迎です。

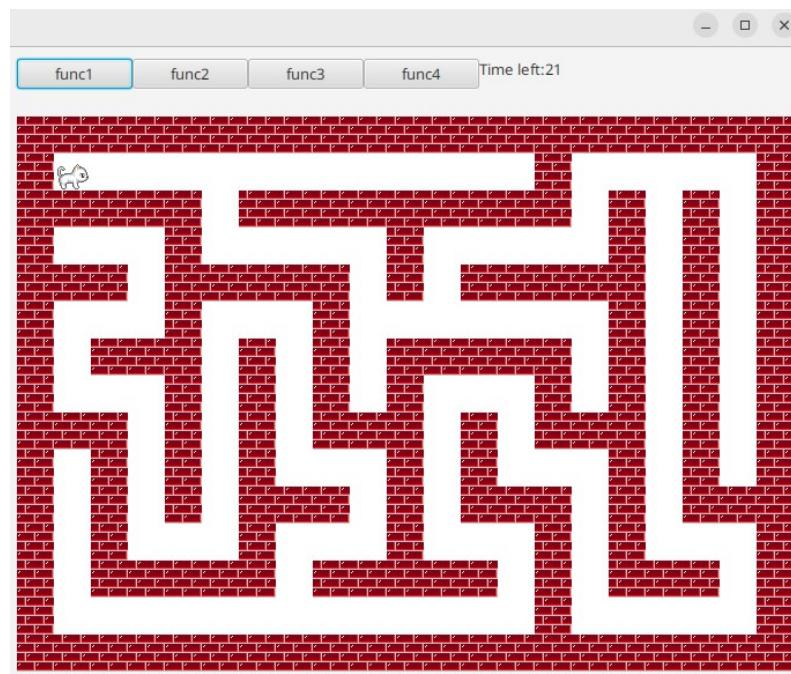
グループの独自性を出すために、他のグループがどのような機能を実装しようとしているかサーベイすることも有効です。また、グループ課題の評価には、発表会での教員・TA・聴講者全員の投票結果が反映されます。プロダクトの完成度のみでなく、発表の質も求められます。発表は 5 分程度ですので、デモのための特別なモード(予め設定した初期値で動作するなど)を実装しておくことも有効です。発表でどのようにアピールするかを検討することも大切です。

¹ Github にグループメンバーが書いたプログラムをアップロードするようにすると、プログラムの差分が自動的に記録されます。以前の版への「巻き戻し」も簡単にできます。zip ファイルを投げ合いながら開発する大変さから解放されます。使用には個人が Github にアカウントを作る必要があります。 <https://github.com>

² Gluon Scene Builder (<https://gluonhq.com/products/scene-builder/>) などがあります。Eclipse (<https://www.eclipse.org/downloads/>) を使用する方法もあります。

ひな型の概要

ゴールに向けて白猫を動かすゲームを、ひな型として提供します。このひな型で用いているのは、以前の課題で学んでいただいた JavaFX (FXML を使用) です。できる限り MVC に則って実装されています。画面のイメージは以下です。白猫がいる部分の正方形を、以下では便宜的に「1マス」と呼びます。

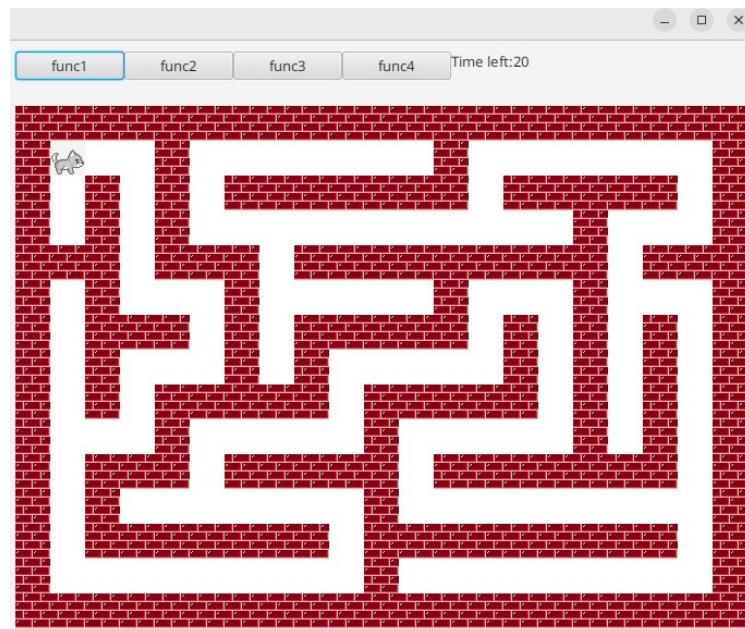


このゲームは以下のようになっています。独自性を出すために変更することは大歓迎です。

- ・ 猫はキーボードの **hjkl** ($\leftarrow \downarrow \uparrow \rightarrow$) で移動する
- ・ 猫は左上 (1, 1) の場所からスタートする ※場所を表す数は 0 から開始しています
- ・ 迷路画面では BGM として 「JoyToTheWorld」 が再生される
- ・ ユーザが **func1** を押すと、BGM が止まり、ゲームオーバー画面が表示される
- ・ “Time left:” の右隣に表示されている数字は、残り時間 (秒) を表す
- ・ 残り時間が 0 になると、BGM が止まり、ゲームオーバー画面が表示される
- ・ ユーザがゲームオーバー画面で「迷路に戻る」を押すと、迷路画面が表示され、BGM が再生される
- ・ “MoveChara.java” にある変数 **useSpriteSheet** の値を **true** にすると (デフォルトは **false**)、png フォルダに保存されているスプライトシートを読み込み、キャラクターを変更できます。デフォルト (**useSpriteSheet=false**) では、バラバラな猫画像を読み込んでゲーム画像に表示しています。次の条件を満たす下図のようなスプライトシートであれば、パスを設定するだけで新しいキャラクターに変更できます：
 - 上から下に、4枚の下向き、左向き、右向き、上向きの画像が並んでいる。
 - 左から右に、3枚のアニメーション画像が並んでいる



犬の画像素材は「ぴょや」 (<https://pipoya.net/sozai/>) のフリー素材を利用しています。



上記以外の独自機能として、例えば以下のような機能が考えられます。

- ・ 猫がゴールに着くとゲームクリアになる
- ・ 猫が迷路上で特定のアイテムを取得すると何かが起こる
- ・ 猫が何かをすると、新しい地図が生成される
- ・ ユーザが func2～func4 のキーを押すとゲーム上で何らかのイベントが発生する

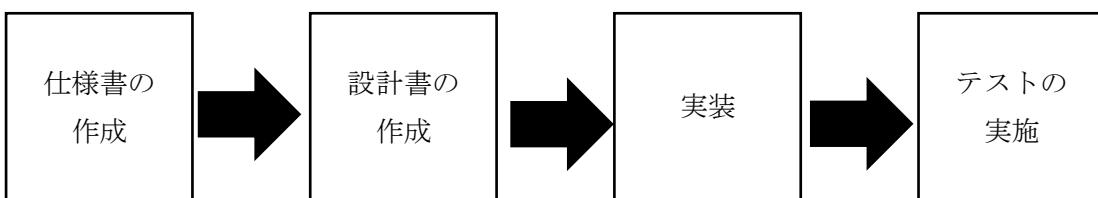
本課題で学習する開発の流れ

実際のシステム開発の世界で行われている仕様書の作成・設計書の作成・実装・テストの実施という一連の仕事をみなさんと体験していただきます。

仕様（策定）・設計・実装・テスト

システム開発の世界では、仕様書の作成・設計書の作成・実装（コーディング）・テストの実施、という作業を通じて、システムを完成させていきます。それぞれの用語の意味は以下の通りです。

- ・ **仕様**：ユーザが「こういう機能があったらいいな」という目線で語ったことを文章や図で表したもの
- ・ **設計**：どのようにして仕様で求められている機能を作るのかを決めること
- ・ **実装**：設計に沿ってプログラムを作ること
- ・ **テスト**：プログラムが期待された通りに動くこと、期待されていないように動かないことを確認すること



今回の課題で求められる仕様・設計・実装・テストの作業内容を以下に示します。

- ・ **仕様**
 - ゲームでできるようにしたい機能を文章や図などで表す。
- ・ **設計**
 - どのような機能を作成するのかについて文章で表現する。いつ、どこへ、何を、どのようにするかが客観的にわかるよう数字を用いて記述するとよい。主語と述語が対応しているかも確認すること。主語は、ユーザ、猫など、述語は○○を押した、○○に触れたなど。図表を用いてもよい。
- ・ **実装**
 - プログラムを作る。また、作成したクラスやメソッド等がどのような機能を持っているのか、他のどのようなクラスやメソッドと、どのように関係しているのかをソースコード内のコメントとして書く。可能であれば javadoc 形式とする。
- ・ **テスト**
 - 設計で定められた入力とは異なるものが入力されても問題ないか、キャラクターに怪しい動きをさせても不具合が生じないか等、イレギュラーなケースをできる限り多く想定し、それらを全て試す。設計にある通りの挙動になるかだけでなく、設計にない動きもテストするのが重要（例：猫が壁を突き抜けて歩くことができないことを確認する）。

設計・実装・テストの例

以下に仕様・設計とテストの例を示します。結構分量が多いことに驚くかもしれません、1つのプロダクトを複数名で協力しながら不具合がないように完成させるには、まだ不十分な部分もあります。実際のプログラムは `MapGameController.java`, `MoveChara.java`などを確認してください。

・ 仕様

- 任意のキーを押すことによって、キャラクターが迷路内を上下左右に移動できるようにしてほしい。

・ 設計

➤ 左へ進む機能

- ❖ ユーザが `h` キーを押したとき、白猫が現在いるマスよりも **1** マス左へ移動する。ただし、左隣のマスが壁の場合は、白猫は移動しない。
- ❖ 左へ移動しようとする際には、白猫が左を向いた画像に切り替える。また、png ディレクトリにある左向きの画像 3 枚を **0.5** 秒おきに切り替えて、白猫が足踏みをしているように見せる。

➤ 右へ進む機能

- ❖ ユーザが `l` キーを押したとき、白猫が現在いるマスよりも **1** マス右へ移動する。ただし、右隣のマスが壁の場合は、白猫は移動しない。
- ❖ 右へ移動しようとする際には、白猫が右を向いた画像に切り替える。また、png ディレクトリにある右向きの画像 3 枚を **0.5** 秒おきに切り替えて、白猫が足踏みをしているように見せる。

➤ 上へ進む機能

- ❖ ユーザが `k` キーを押したとき、白猫が現在いるマスよりも **1** マス上へ移動する。ただし、上隣のマスが壁の場合は、白猫は移動しない。
- ❖ 上へ移動しようとする際には、白猫が上を向いた画像に切り替える。また、png ディレクトリにある上向きの画像 3 枚を **0.5** 秒おきに切り替えて、白猫が足踏みをしているように見せる。

➤ 下へ進む機能

- ❖ ユーザが `j` キーを押したとき、白猫が現在いるマスよりも **1** マス下へ移動する。ただし、下隣のマスが壁の場合は、白猫は移動しない。
- ❖ 下へ移動しようとする際には、白猫が下を向いた画像に切り替える。また、png ディレクトリにある下向きの画像 3 枚を **0.5** 秒おきに切り替えて、白猫が足踏みをしているように見せる。

・ テスト

➤ 左へ進む機能

- ❖ 白猫が現在いるマスの左隣のマスが空白のとき、ユーザが `h` キーを押したら、白猫が現在いるマスよりも **1** マス左へ移動することを確認する。
- ❖ 白猫が現在いるマスの左隣が壁のとき、ユーザが `h` キーを押しても白猫がどこへも移動しないことを確認する。
- ❖ 左へ移動しようとする際に、白猫が左を向いた画像が表示されていることを確認する。
- ❖ 左へ移動しようとする際に、白猫が左を向いた画像が **0.5** 秒間隔で 3 枚を切り替えて表示されていることを確認する。

➤ 右へ進む機能

- ❖ 白猫が現在いるマスの右隣のマスが空白のとき、ユーザが `l` キーを押したら、白猫が現在いるマスよりも **1** マス右へ移動することを確認する。
- ❖ 白猫が現在いるマスの右隣が壁のとき、ユーザが `l` キーを押しても白猫がどこへも移動しないことを確認する。

- ❖ 右へ移動しようとする際に、白猫が右を向いた画像が表示されていることを確認する。
- ❖ 右へ移動しようとする際に、白猫が右を向いた画像が 0.5 秒間隔で 3 枚を切り替えて表示されていることを確認する。
- 上へ進む機能
 - ❖ 白猫が現在いるマスの上隣のマスが空白のとき、ユーザが k キーを押したら、白猫が現在いるマスよりも 1 マス上へ移動することを確認する。
 - ❖ 白猫が現在いるマスの上隣が壁のとき、ユーザが k キーを押しても白猫がどこへも移動しないことを確認する。
 - ❖ 上へ移動しようとする際に、白猫が上を向いた画像が表示されていることを確認する。
 - ❖ 上へ移動しようとする際に、白猫が上を向いた画像が 0.5 秒間隔で 3 枚を切り替えて表示されていることを確認する。
- 下へ進む機能
 - ❖ 白猫が現在いるマスの下隣のマスが空白のとき、ユーザが j キーを押したら、白猫が現在いるマスよりも 1 マス下へ移動することを確認する。
 - ❖ 白猫が現在いるマスの下隣が壁のとき、ユーザが j キーを押しても白猫がどこへも移動しないことを確認する。
 - ❖ 下へ移動しようとする際に、白猫が下を向いた画像が表示されていることを確認する。
 - ❖ 下へ移動しようとする際に、白猫が下を向いた画像が 0.5 秒間隔で 3 枚を切り替えて表示していることを確認する。

本課題の必須機能の仕様

以下に、グループで実現してもらいたい 5 つの必須機能の仕様を示します。ユーザの希望を仕様からくみ取つて、設計を行い、設計を参考にして実装し、実装したものをテストしてください。なお、必須機能間で情報交換をしなければうまく進められないところが数多くあります。情報交換を行って機能間の整合性を取るところまでが担当者の仕事です。

必須機能 1：BGM と SE の設定

仕様：ゲーム中はゲームに合った BGM を流してほしい。また、ゲームオーバーの時にはゲームオーバーらしい BGM を流してほしい。BGM は著作権などの問題にひつかからないものにしてほしい。また、キャラクターの動きに合わせた効果音(SE)を 2 種類以上含めてほしい。

ヒント：`StageDB.java` にある `getMainSound()` を例にして、`getGameOverSound()` を実装する。BGM ファイルは `contest` フォルダ内にある `sound` フォルダに入れる。

必須機能 2：キャラクターの置き換え&アイテム作り

仕様①：キャラクターを白猫以外の何かにしてほしい。

仕様②：マップ上に配置するためのアイテムの設定を決めてほしい。アイテム画像も用意してほしい。アイテム画像は著作権などの問題にひつからないものにしてほしい。

ヒント①：`MoveChara.java` でキャラクター画像を設定する。必要に応じてキャラクターのアニメーション設定も変更する。アニメーション関連の設定は `MoveChara.java` の中にある「class ImageAnimation」を参照。

ヒント②：アイテム画像は 32x32 ピクセルとする。`MoveChara.java` にある関数 `resizeTo32` が参考になる。画像は `contest` フォルダ内にある `png` フォルダに入れる。画像ファイル名は半角英数字にする。画像ファイル名を、必須機能 3 担当者に伝える。

必須機能 3：新マップの生成

仕様：`func2` を押したら、新しくランダムに作られたマップ画面が表示されるようにしてほしい。

ヒント：`MapData.java` は、ゲームが起動するたびに呼びだされている。`MapData.java` の内部には、まず壁で埋め尽くされた迷路の元を作り(`fillMap` メソッド)，その後そこへランダムに通路を作る命令がある(`digMap` メソッド)。`MapData.java` を呼び出しているのは、`MapGameController.java` にある `initialize` メソッドの「`mapData = new MapData(21, 15);`」。

必須機能 4：アイテムの配置

仕様：キャラクターがゲットできるアイテムを複数、迷路に置いてほしい。また、アイテムをゲットするとゲーム上で何かが起こってほしい。

ヒント：地図を描く命令を出しているのは `MapGameController.java` の `drawMap` メソッド。アイテム画像は必須機能 2 担当の人が作る。アイテムのファイル名もその人から教えてもらう。

必須機能 5：ゴールの作成

仕様：キャラクターがゴールの場所へ行くと、ターミナルにゲームクリアの文字が表示されるようにしてほしい。

ヒント：`MoveChara.java` にある `move` というメソッドが、キャラクターを動かす処理をしている。

`MoveChara.java` は、`MapGameController.java` から呼び出されている。もしキャラクターの現在地が、ゲームで設定したゴール座標と同じ位置だったならば、ゲームクリアの文字を表示するようにする。

ひな型の実行方法

MS Teams から「**contest.zip**」をダウンロードし, **unzip** コマンドで展開してください。

```
$ unzip contest.zip
```

「**contest**」という新しくできたフォルダへ移動してコンパイル・実行し, 動作を確認してください。

```
$ cd contest
$ javafxc *.java
$ javafx MapGame
```

今後のスケジュール

以下は、最終期限までに最低限ここまでやっておいたほうがよい、というスケジュールです。実際には、もう少し早く進めることを勧めます。1月中旬までに全体を作り終えるような気持ちで進めてください。個別のプログラムはうまく動いても、結合するとうまく動かなくなる事件（悪夢）が毎年頻発しています。

1.準備（12/12）

今回やることはメンバー同士の自己紹介、グループ内での連絡手段確立、グループ内での課題内容の把握、グループ内での計画立案と分担の決定（提出担当者表と分担表を代表者が提出する）、です。

5.の最終発表会を除いて、1.準備から6.発表会後まで計5回の課題提出担当者の分担表を作成し、代表者1名が提出してください（レポートひな型はTeamsの「XXグループ_1212レポート.docx」XXはグループ番号で置き換えてください）。各回の代表者は提出業務の担当者ですので、必要に応じてグループメンバーに催促し、提出できる状態まで取りまとめを行ってください。本日も含めると計5回、グループで課題提出を行う機会があります。つまり、1人につき1回程度、提出担当回がまわってきます。

＜提出担当者表＞

スケジュール	グループ課題提出者（例）
1.準備（12/12）	Aさん
2.構想発表（12/19）	Bさん
3.必須機能のみのゲームの提出（1/9）	Cさん
4.独自機能設計・テストの提出（1/16）	Dさん
6.発表会後（1/23）	Eさん

また、各機能の仕様・設計・実装・テストを誰が担当するのかを決めて以下の分担表に記入し、代表者が提出してください。本課題でグループメンバーは、**1人1つ以上の機能を実装**していただきます。実装以外にも、**仕様策定・設計とテスト**という仕事があります。各機能における仕様策定・設計・実装・テストは、作者以外の新鮮な目でチェックするという目的を果たすために、作者とは別の人気が担当しなければなりません。面倒なようではありますが、1つの機能を完成させる過程に複数の人を関与させることで、1人では気付かなかつた勘違いを取り除いたり、1人では気付かなかつたより良い実現の方法に気付いたりするという協調の利点を生かすためです。システム開発現場でもこのように、異なる人が異なる観点から同じ機能を確認することが多々あります。例えば以下のように分担する方法が考えられます。独自機能は、必須機能や他のグループと異なる機能を最低2つ考えて実現していただきます。

＜分担表＞

機能名	仕様	設計	実装	テスト
必須機能1		Aさん	Eさん	Bさん
必須機能2		Bさん	Aさん	Cさん
必須機能3		Cさん	Bさん	Aさん
必須機能4		Dさん	Cさん	Bさん
必須機能5		Eさん	Dさん	Cさん
独自機能1	Aさん	Cさん	Eさん	Dさん
独自機能2	Bさん	Dさん	Aさん	Eさん

2.構想発表

12/19までに、これまでに実装し終えた必須機能の紹介、今後製作予定の独自機能の仕様の紹介、現在悩んでいること・困っていることを紹介してください。

また、各必須機能の仕様・設計・テストについて日本語で記述したものも提出してください。

3.必須機能のみのゲームの提出

1/9までに、必須機能5つを結合してゲームとしてほぼ動くようになったもの（プログラムと `xml`）をグループで1セット提出してください。まだできていない部分がある場合は、何ができないのかを具体的に報告してください。

4.独自機能の設計・テストの提出

1/16までに、2つ以上の独自機能の設計とテストを日本語で記述したものを、グループで1セット提出してください。構想発表を踏まえて方針変更することも可能とします。まだできていない部分がある場合は、何ができるないのかを具体的に報告してください。

5.最終発表会

1/23の最終発表会では、作成したゲームの特徴が際立つようなデモを行ってください。また、工夫した点などを説明してください。さらに、質問に応答してください。発表者と質疑応答担当者は当日ランダムに決めます。発表順も当日決めます。したがって授業開始時点から全員が参加していること、全員が発表できるように準備しておく必要があります。発表の際に画面共有の操作を手伝う等は、発表者以外の人が行ってかまいません。

6.発表会後

発表会終了後は、発表会当日に得られた情報も踏まえて、個人およびグループでレポートを作成し、グループでとりまとめて提出してください。作品も提出していただきます。内容は本資料末尾を参考にしてください。

注意事項

画像や音楽などの素材はどのようなものを使用してもかまいません。ただし、素材を提供している人が示している利用規約を遵守してください。たとえば「いらすとや」では、規約の範囲内であれば個人・法人、商用・非商用問わず無料で素材を使用することができますが、規約の範囲外であれば著作者に問い合わせをする必要があります。参考（いらすとや利用規約） <https://www.irasutoya.com/p/terms.html>

どんな場合も、他の方の著作物を利用するときには利用規約を確認してください。無料で使用することができる素材であっても、利用範囲や利用目的は作者の意図によって異なる点です。場合によっては、著作物の出処情報を併記する必要があったり、特定の目的に限って使用しなければならなかったりします。使い方によっては使用者側の責任が問われる場合があります。

もし著作物の使用に関して、利用規約を読んでも迷うことがあれば、教員やTAへ連絡してください。本グループ課題では、学校での教育目的で使用すること、特定された相手のみに閉じた状態で公開されること、私的・非商用利用であることがポイントになります。

なお、無料で提供されているアプリなどのプログラムについても、ライセンス事項が定められています。Ubuntu や JDK にも定められています。

今後の提出物

以下は期末レポートの予告です。みなさんは問題ないと思いますが、グループ課題に関する記述で虚偽申告が

あった場合にも**不正行為**とみなされますので、ご注意ください。

個人レポート（最終発表会後）

- ・ グループで自分以外の各メンバーがグループ課題の完成に向けてどのようなことをしたのか、1人ひとりについて記しましょう。代表して先生に質問した、グループメンバー間のケンカを仲裁した、話し合いの日時を調整した、など幅広く考えてください。
- ・ 他のグループの発表会を聴いて疑問に思ったこと、素晴らしいと思ったことについて、それぞれ最低1点ずつ記しましょう。
- ・ 自分達のグループの進め方を振り返り、どのような点を改善すると今後もっとよい進め方ができると考えられるかを記しましょう。

以上について書いたものをグループの代表者へ提出し、代表者1名から教員に提出していただきます。

グループレポート（最終発表会後、分担して執筆したものを代表者1名が提出）

- ・ 作品のプログラムと `xml`
- ・ どのような独自機能を作成したかの説明
- ・ 5つの必須機能と2つ以上の独自機能についての役割分担表

※以下は例。12月時点から変更があれば、変更箇所がわかるように記載する。

機能名	仕様	設計	実装	テスト
必須機能 1		Aさん	Eさん	Bさん
必須機能 2		Bさん	Aさん	Cさん
必須機能 3		Cさん	Bさん	Aさん
必須機能 4		Dさん	Cさん	Bさん
必須機能 5		Eさん	Dさん	Cさん
独自機能 1	Aさん	Cさん	Eさん	Dさん
独自機能 2	Bさん	Dさん	Aさん	Eさん

- ・ 5つの必須機能と2つ以上の独自機能について作成した仕様の文章、プログラム処理の内容を説明したもの（クラスやメソッドを作った場合はその名前も明記する）、テスト項目の文章
- ・ 実行結果
 - PNG形式の画像と、このゲームで遊ぼうとする人（大学生程度を想定）へのごく簡単な事前説明
- ・ グループメンバーの個人レポート
 - メンバー全員から集めた個人レポートをそのまま貼り付ける
- ・ 参考文献
 - 今回の課題を進めるに当たって参考にした書籍、webサイトのURL、誰かの発言などを列挙
- ・ 謝辞
 - グループ以外の人で、課題達成に特に協力してくれた人、団体等があれば明記